Learning for Crowdsourcing: Online Dispatch for Video Analytics with Guarantee

Yu Chen†, Sheng Zhang*†, Yibo Jin*†, Zhuzhong Qian†, Mingjun Xiao§, Ning Chen†, and Zhi Ma†
†State Key Lab. for Novel Software Technology, Nanjing University, P.R. China
§School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, P.R. China

***Abstract*—Crowdsourcing enables a paradigm to conduct the manual annotation and the analytics by those recruited workers, with their rewards relevant to the quality of the results. Existing dispatchers fail to capture the resource-quality trade-off for video analytics, since the configurations supported by various workers are different, and the workers' availability is essentially dynamic. To determine the most suitable configurations as well as workers for video analytics, we formulate a non-linear mixed program in a long-term scope, maximizing the profit for the crowdsourcing platform. Based on previous results under various configurations and workers, we design an algorithm via a series of subproblems to decide the configurations adaptively upon the prediction of the worker rewards. Such prediction is based on volatile multi-armed bandit to capture the workers' availability and stochastic changes on resource uses. Via rigorous proof, the regret is ensured upon the Lyapunov optimization and the bandit, measuring the gap between the online decisions and the offline optimum. Extensive trace-driven experiments show that our algorithm improves the platform profit by 37%, compared with other algorithms.**

I. Introduction

The mobile crowdsourcing [1–5] has become an increasingly promising paradigm, which leverages the resource of the workers, recruited from mobile users, to conduct various crowdsourcing jobs, including manual annotations [6], image labelling [7] and further analytics [8], with the rewards highly depending on the quality of the results. For the video analytics with manual annotations or image labelling, the configurations need to be determined carefully, since the accuracy of object detection [9] relies on them, including frame rates, resolutions, etc. As a result, the dispatcher of the crowdsourcing needs to choose the most suitable configurations with maximum profit.

Unfortunately, optimally determining the configurations for videos over recruited workers, as in Fig. 1, with their availability changing over time, faces multiple challenges as follows:

First and foremost, the configurations supported by various recruited workers are quite different. Limited to the computing capacity and the hardware [10], some workers fail to support the configurations with heavy resource involved. Furthermore, after deploying a specific video analytics model for a worker, the configurations supported by such a model [11–13] are often limited, since the video analytics models often have their own input formats [14–16]. And the fixed formats restrict the inputs of the video analytics, e.g., requiring those videos with a fixed range of resolutions. When deciding the configurations and the workers for the video analytics, these relationships should be carefully considered, in case the desired configurations fail to be supported on the workers, especially when the availability of workers is changing over time dynamically.

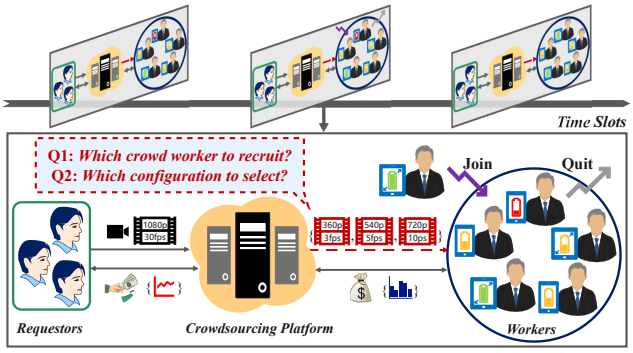


Fig. 1. Illustration of Mobile Crowdsourcing for Video Analytics

Second, the resource-quality trade-off for the videos essentially affects the rewards and it needs to be determined before the execution. Although the configurations to obtain the high-quality results for video analytics are desired, they often incur heavy computation on workers with more resource consumed [17]. Meanwhile, the workers request a higher reward from the crowdsourcing platform due to their contributions [18]. To balance the profit and the quality of results, the platform has to decide the resource-quality trade-off. Unfortunately, the trade-off must be conducted before the execution, because it is part of the input [19]. Furthermore, the trade-off is highly relevant to videos' inner contents. It is improper to adopt the previous one directly for a new video, although the feedback from the previous executions [20] under various configurations potentially helps the trade-off decisions for video analytics.

Third, the resources consumed on workers in terms of both transmission and computation are uncertain. As shown in our case studies later, although all settings for the execution of the video analytics are fixed, the energy consumed by devices changes over time [21]. Similarly, even if all settings are fixed for the transmission, the energy consumed changes over time as well [22]. Such stochastic inner changes on the resource consumption hampers the workers to estimate the rewards and also hampers the platform to determine the configurations and workers in advance for video analytics with maximum profit. Although previous observable inputs actually help to capture the changes of those stochastic inputs, the availability of the

*The Corresponding Authors are Sheng Zhang (sheng@nju.edu.cn) and Yibo Jin (yibo.jin@smail.nju.edu.cn). This work was supported in part by NSFC (61872175, 61832008), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

(a) Detect with Low Resolution (b) Detect with Low Frame Rate

Fig. 2. Inappropriate Configurations for Detecting Objects

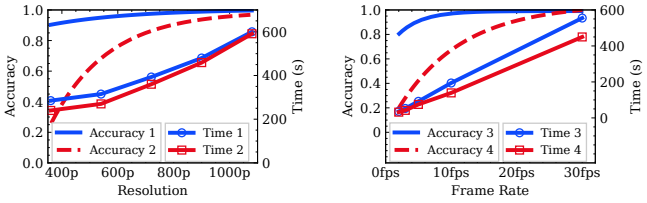


Fig. 3. Trade-off under Various Resolutions and Frame Rates

workers dynamically changing over time further makes it hard to estimate the resource consumptions on workers.

Existing works fall insufficient for tackling these previous challenges. Some works [23–25] have considered the configurations for videos, but they fail to optimize the crowdsourcing profit. Other works [26–30] have studied the worker selections, but they fail to decide the configurations. The rest [31–35] have discussed the resource changes, but they fail to treat them with guarantee under dynamic workers' availability.

In this paper, to overcome previous challenges for crowdsourcing, we formulate a non-linear mixed program with the objective of maximum profit for the platform in a long-term scope, capturing the dynamic availability of recruited workers. Essentially, the profit depends on the quality of the analytics results, and the energy consumption on both transmission and computation is considered. Furthermore, in order to ensure the quality, a guarantee is also promised by the platform.

We then design an online algorithm via a series of subproblems solved over time. These subproblems take the previous analytics results under various configurations and workers into consideration, and are willing to dispatch those video analytics tasks to the workers with minimum resource consumption, and to configure the most suitable settings. Such subproblems are based on the prediction regarding the resource consumption on recruited workers. In order to capture the stochastic resource usage, also catering to the dynamic workers' availability, the volatile multi-armed bandit is used for the prediction. By using these predictive inputs, these subproblems further balance the cumulative violation regarding the platform guarantee and the profit incurred within current time slot. Via rigorous proofs, by using our designed algorithm with a series of subproblems solved upon the prediction, the regret is ensured based on the Lyapunov optimization and the bandit. Such regret measures the gap between the results from our online algorithm and the offline optimum. Furthermore, the guarantee promised by the crowdsourcing platform for video analytics is ensured.

Extensive trace-driven experiments with the videos derived from the AI City Datasets 2019 [36] confirm the superiority of our online dispatcher for video analytics upon YOLOv3 [11], compared with other algorithms. The profit increases 37.9% on average with 0.2‰ violation on promised guarantee.

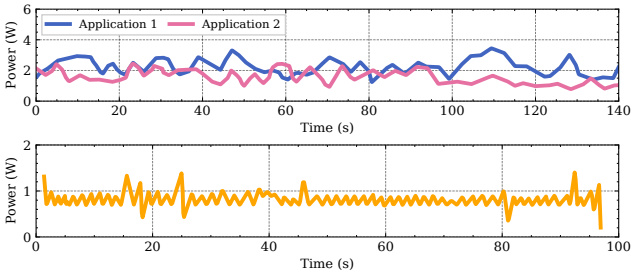


Fig. 4. Changes on Computation (Top) and Transmission (Bottom)

II. System Model

A. Preliminary Case Studies for Video Crowdsourcing

Video Configurations: As illustrated in Fig. 2(a), when the background is similar to objects and the objects are small, detecting them precisely often requires a higher resolution [14]. As shown in Fig. 3, with the growth of the resolutions, the accuracy increases and the detection time also grows dramatically. As illustrated in Fig. 2(b) [16], when the objects are moving fast in the video, detecting them precisely from the videos requires a higher frame rate. As shown in Fig. 3, with the growth of the frame rates, the accuracy improves, but the detection time increases as well [19]. Thus, for different video analytics, the configuration should be adapted to its contents for better detection accuracy. However, more detection time and corresponding resources have to be then involved.

Resource on Workers: As illustrated in Fig. 4, the energy consumption incurred by both computation and transmission changes over time [21, 22] on the devices. Furthermore, for different computation workloads, the overall energy consumptions also vary. Similarly, for different video analytics under various configurations on workers, the workloads incurred are different, and the resource consumption also changes over time, even if the configurations are fixed. Such stochastic inner changes on the resource usage hamper us from precisely estimation on resource consumption, and further hamper the platform from precise estimation on the profit [18].

Opportunity and Challenges: Although adaptively preparing the best configurations for videos improves the quality of analytics, the rewards for the workers depend on consumed resources. Futher, the stochastic changes on resources hamper us from precisely prediction on the consumption, which also hamper the platform from profit estimation.

B. System Settings and Models

We summarize the major notations used in Table I.

Mobile Crowdsourcing: We consider the crowdsourcing system, with the overall time scope being divided into many time slots, denoted as $\mathcal{T} = \{1, ..., T\}$. The platform receives the video analytics from the requestors and dispatches the tasks split from videos to recruited workers. Within each time slot, some new devices may be recruited as the workers and the old ones may terminate due to low volume of battery. We further divide each time slot into multiple epochs $\mathcal{J}^t$, and the available worker set in epoch j, slot t is denoted as $\mathcal{N}_j^t$.

Video Analytics: For each time slot, the platform needs to determine the worker selection and configuration decisions for M^t tasks, the set of which are $\mathcal{M}^t = \{s_1^t, s_2^t, ..., s_{M^t}^t\}$.

TABLE I
MAJOR NOTATIONS USED FOR MODEL

Symbols	Descriptions
$\boldsymbol{\mathcal{N}}_j^t$	Set of workers in epoch j time slot t
$\boldsymbol{\mathcal{M}}^t$	$\{s_1^t, s_2^t, ..., s_{M^t}^t\}$, set of video analytics tasks in time slot t
$\boldsymbol{\mathcal{K}}_m^t$	$\{s_{m,k}^t \vert k = 1, 2, ..., K_m^t\}$, set of subtasks in task s_m^t
a^t	Average analytics accuracy for all subtasks in time slot t
e_n^t	Energy consumption on worker n's device in time slot t
U^t	Crowdsourcing profit in time slot t
F_m^t	Original frame rate of to-be-analyzed video in task s_m^t
A^{min}	Accuracy guarantee promised by crowdsourcing platform
$x_{m,k,n}^t$	Whether to allocate subtask $s_{m,k}^t$ to worker n
$f_{m,k}^t$	Frame rate selected for subtask $s_{m,k}^t$

Following the ideas in [37–39], each video analytics task is relatively large and can be divided into multiple subtasks (e.g., several frames for each subtask, which is suitable for the execution on mobile devices). Let K_m^t be the number of subtasks in task s_m^t, and the subtask set is $\boldsymbol{\mathcal{K}}_m^t = \{s_{m,k}^t | k = 1, 2, ..., K_m^t\}$, where $s_{m,k}^t$ is the k-th subtask of task s_m^t.

Analytics on Workers: Considering the limited capacity of mobile devices [10], each worker's mobile device is equipped with one CNN model for analytical simplicity, of which the input resolution is r_i. Nevertheless, our proposed model can also handle multiple CNNs on each device. Once a worker is selected, the resolution is determined as the input resolution of the CNN model on the worker's mobile device. We assume that the original resolution of each to-be-analyzed video is high, and it can be reduced to the selected resolution by simple down-sampling [40, 41]. We use $x_{m,k,n}^t$ to represent whether a subtask $s_{m,k}^t$ is allocated to worker n, and $f_{m,k}^t$ represents the frame rate selected for subtask $s_{m,k}^t$. Therefore, in each time slot t, the mobile crowdsourcing platform needs to determine the worker selection and frame rate adaptation for each subtask to maximize the crowdsourcing profit subject to the time-averaged accuracy guarantee in the long term.

Accuracy of Analytics: Profiling the relationship between the configurations and the accuracy is non-trivial, since the configuration is multi-dimensional, including resolution and frame rate. Different decision variables can affect the accuracy in different ways. Based on existing prior studies [9, 14, 38] and the performance measurements obtained from our preliminary case studies, we have two important observations: i) the relationship between accuracy and resolution/frame rate can be formulated as a concave function; ii) frame resolution adaptation and frame sampling rate impact accuracy independently. Based on the above two observations, the accuracy of subtask $s_{m,k}^t$ in epoch j, slot t can be calculated as

$$a_{m,k}^t = \epsilon_m^t(\sum\nolimits_{n=1}^{|\boldsymbol{\mathcal{N}}_j^t|} x_{m,k,n}^t r_n)\phi_m^t(f_{m,k}^t),$$

where the concave functions $\epsilon_m^t(r)$ and $\phi_m^t(f)$ represent the accuracies with respect to resolution r and frame rate f for task s_m^t, respectively, and $\sum_n x_{m,k,n}^t r_n$ is the selected frame resolution for task s_m^t. Therefore, it is easy to get the average analytics accuracy for all subtasks in time slot t as

$$a^t = \sum\nolimits_{m=1}^{M^t} \sum\nolimits_{k=1}^{K_m^t} a_{m,k}^t / \sum\nolimits_{m=1}^{M^t} \sum\nolimits_{k=1}^{K_m^t}.$$

Consumptions of Analytics: For the workers, the battery becomes one of the most concerns because it is inconvenient to recharge the devices. The energy consumption mainly consists of two parts: transmission energy and processing energy.

The transmission energy consumption is caused by worker's downloading video data from the mobile crowdsourcing platform, which is proportional to the size of downloaded video data [14]. Following the idea in [42], the data size of a video frame with resolution r can be calculated as αr^2 bits, where α is a constant. Thus, we get the energy consumption for worker n's downloading video data in time slot t as

$$e_n^{d,t} = \sum\nolimits_{m=1}^{M^t} \sum\nolimits_{k=1}^{K_m^t} \gamma_{m,n}^t \alpha (x_{m,k,n}^t r_n)^2 f_{m,k}^t,$$

where $\gamma_{m,n}^t$ denotes the energy consumption for downloading one bit of the input for video analytics on worker n's device.

The processing energy consumption is resulted from local video frame processing on worker's device. We use $\mu_{m,n}^t$ as the energy consumption for processing one frame on worker n's device [43], then the processing energy consumption on worker n's mobile device in time slot t is calculated as

$$e_n^{c,t} = \sum\nolimits_{m=1}^{M^t} \sum\nolimits_{k=1}^{K_m^t} \mu_{m,n}^t x_{m,k,n}^t f_{m,k}^t.$$

The overall energy consumption on worker n in time slot t is $e_n^t = e_n^{d,t} + e_n^{c,t}$, where $\gamma_{m,n}^t$ and $\mu_{m,n}^t$ are stochastic.

Profit of Crowdsourcing: For a task requestor, the accuracy of video analytics is of considerable importance. Similar to existing works [38, 44], we use a concave function $G_m^t(a)$ to model the revenue function, with respect to the accuracy a for task s_m^t. Thus, the crowdsourcing revenue received from requestors in time slot t can be calculated as

$$I^t = \sum\nolimits_{m=1}^{M^t} \sum\nolimits_{k=1}^{K_m^t} G_m^t(a_{m,k}^t).$$

Since the energy consumption plays an important role in the mobile crowdsourcing, the payment to crowd workers mainly covers the execution cost of energy consumption. Let ω_n denote the payment, for consuming each unit of energy, priced by worker n [18]. Thus, we calculate the total monetary cost of crowdsourcing platform in time slot t as

$$O^t = \sum\nolimits_{j=1}^{|\boldsymbol{\mathcal{T}}^t|} \sum\nolimits_{n=1}^{|\boldsymbol{\mathcal{N}}_j^t|} \omega_n e_n^t,$$

The profit (a.k.a utility) of the crowdsourcing platform can be calculated as the revenue received from the task requestors minus the payment to the crowd workers [18] involved. Thus, the crowdsourcing profit in time slot t is $U^t = I^t - O^t$.

C. Problem Formulation

For crowdsourcing platform, the objective is to maximize the profit under the long-term accuracy constraint. Based on the above models, our considered problem is formulated as

$$\mathbb{P}_1 : \max_{x,f} \lim_{T\to+\infty} \frac{1}{T} \sum\nolimits_{t=0}^{T} U^t \tag{1a}$$

$$s.t.\ x_{m,k,n}^t = \{0,1\}, \forall t, \forall m \in \boldsymbol{\mathcal{M}}^t, \forall k \in \boldsymbol{\mathcal{K}}_m^t, \forall n \in \cup_j \boldsymbol{\mathcal{N}}_j^t, \tag{1b}$$

$$\sum\nolimits_{j=1}^{|\boldsymbol{\mathcal{T}}^t|} \sum\nolimits_{n=1}^{|\boldsymbol{\mathcal{N}}_j^t|} x_{m,k,n}^t = 1, \forall t, \forall m \in \boldsymbol{\mathcal{M}}^t, \forall k \in \boldsymbol{\mathcal{K}}_m^t, \tag{1c}$$

$$1 \le f_{m,k}^t \le F_m^t, \forall t, \forall m \in \boldsymbol{\mathcal{M}}^t, \forall k \in \boldsymbol{\mathcal{K}}_m^t, \tag{1d}$$

$$\lim_{T\to+\infty} \frac{1}{T} \sum\nolimits_{t=0}^{T} a^t \ge A^{min}. \tag{1e}$$

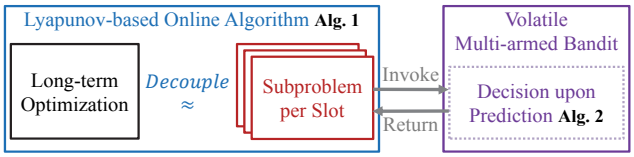


Fig. 5. Volatile Multi-armed Bandit based Lyapunov Optimization

Constraints (1b) and (1c) ensure that each subtask is allocated to one and only one worker per time slot t. Constraint (1d) ensures that the selected frame rate cannot exceed F_m^t, the original frame rate of to-be-analyzed video in task s_m^t. The last constraint (1e) requires that the time-averaged analytics accuracy should satisfy the accuracy guarantee A^{min}.

The difficulty in obtaining the optimal solution to problem $\mathbb{P}_1$ is the lack of future information. In order to optimally solve $\mathbb{P}_1$, near future information about varying video content and its corresponding best configuration is needed, which is hard to predict precisely in advance. In addition, $\mathbb{P}_1$ is a mixed integer nonlinear programming problem, and it is very difficult to tackle even if the future information can be known a priori. An online approach is needed to efficiently determine the worker selection and configuration adaptation for the video analytics tasks on the fly without foreseeing the future.

III. Online Algorithm Design

To decouple the long-term constraint, we use the Lyapunov optimization to convert the original time-averaged problem $\mathbb{P}_1$ into a series of subproblems per slot, as shown in Fig. 5. Upon invoking **Algorithm 2** per time slot using multi-armed bandit, a series of subproblems regarding the video analytics over workers are actually controlled in **Algorithm 1**.

A. Lyapunov-based Online Optimization

The major difficulty of directly solving $\mathbb{P}_1$ is that the long-term analysis accuracy constraint (1e) couples the worker selection and frame rate decisions across different time slots. To address this difficulty, we apply the Lyapunov optimization and define a virtual accuracy deficit queue to guide the decisions, which follows the long-term analysis accuracy constraint. Specifically, we assume that the initial queue backlog $q(0)$ is 0, and the queue length evolves as follows:

$$q(t+1) = [q(t) + A^{min} - a^t]^+, \tag{2}$$

where $[\cdot]^+$ denotes $\max\{\cdot, 0\}$, and queue backlog $q(t)$ indicates the deviation of current accuracy from the accuracy constraint in time slot t. In order to get a higher profit, if the platform attempts to simply reduce the expenditure O^t by lowering the frame rate, the queue backlog $q(t)$ will increase unboundedly, resulting in the unacceptable accuracy. To keep the accuracy queue stable, we define the quadratic Lyapunov function as $L(q(t)) = \frac{1}{2}(q(t))^2$, which represents the congestion level in accuracy deficit queue. When the value of $L(q(t))$ is small, it implies that the queue backlog is also small and the accuracy queue has strong stability. In order to keep the queue stability and enforce the accuracy constraint by persistently pushing the Lyapunov function to a lower value, we introduce the one-slot Lyapunov drift:

$$\Delta(q(t)) = \mathbb{E}[L(q(t+1)) - L(q(t))|q(t)], \tag{3}$$

Algorithm 1: Lyapunov Online aLgorithm (LOL)

Input: $q(0) \leftarrow 0$, A^{min}, ω_n, F_m^t, G_m^t, $\forall t$

for $t = 0$ *to* T **do**
  Profile accuracy model functions ϵ_m^t and ϕ_m^t, $\forall m$;
  for $j = 1$ *to* $|\mathcal{J}^t|$ **do**
    $\mathcal{L}_m^t \leftarrow$ Set of subtasks left in s_m^t, $\forall m$;
    for *Subtask* $s_{m,k}^t$ *in* $\mathcal{L}_m^t$, $\forall m$ **do**
      Prepare $z_{m,k}^t()$ with pending inputs $<\mu, \gamma>$;
      Call **Algorithm 2** and obtain $z_{m,k}^t$;
  Calculate a^t and $q(t+1) \leftarrow [q(t) + A^{min} - a^t]^+$;

which represents the expected change in the Lyapunov function over one time slot when $q(t)$ is given. We have

$$\Delta(q(t)) = \frac{1}{2}\mathbb{E}[q^2(t+1) - q^2(t)|q(t)] \tag{4a}$$
$$\leq \frac{1}{2}\mathbb{E}[(q(t) + A^{min} - a^t)^2 - q^2(t)|q(t)] \tag{4b}$$
$$= \frac{1}{2}(A^{min} - a^t)^2 + q(t)\mathbb{E}[(A^{min} - a^t)|q(t)] \tag{4c}$$
$$\leq B + q(t)\mathbb{E}[(A^{min} - a^t)|q(t)], \tag{4d}$$

where $B = \frac{1}{2}(A^{min} - a^{min})^2$ is a constant value, and $a^{min} = \min_{t\in T}\{a^t\}$ represents the lowest average accuracy in all time slots. The previous inequality comes from the Eq. (2).

We incorporate the accuracy queue stability into the optimized objective function of crowdsourcing profit and define the Lyapunov drift-plus-penalty term as

$$\Delta(q(t)) - V \cdot \mathbb{E}[U^t|q(t)], \tag{5}$$

where we use the positive parameter V to adjust the tradeoff between analysis accuracy improvement and crowdsourcing profit maximization. Plugging (4d) into (5), we have

$$\begin{aligned}&\Delta(q(t)) - V \cdot \mathbb{E}[U^t|q(t)] \leq \\ &\quad B + q(t) \cdot \mathbb{E}[A^{min} - a^t|q(t)] - V \cdot \mathbb{E}[U^t|q(t)].\end{aligned} \tag{6}$$

Decoupling: Rather than directly minimizing the Lyapunov drift-plus-penalty term (5) in each slot, we learn from the min-drift-plus-penalty algorithm [45] in Lyapunov optimization framework and attempt to minimizing the upper bound for the Lyapunov drift-plus-penalty term in each time slot t, i.e.,

$$\mathbb{P}_2 : \max_{x,f} \; q(t) \cdot a^t + V \cdot U^t \tag{7a}$$
$$s.t. \;\; (1b),(1c),(1d). \tag{7b}$$

Notice that solving $\mathbb{P}_2$ requires only the currently available inputs. Considering the additional term $q(t) \cdot a^t$, the crowdsourcing system takes into account the analysis accuracy in the current time slot. When the value of $q(t)$ is large, it implies that minimizing the accuracy deficit is more critical. The accuracy queue is maintained without knowing future information. It guides the worker selection and frame rate adaptation to follow the long-term accuracy constraint, thereby enabling online decision making. The optimized objective function in $\mathbb{P}_2$, i.e., $\sum_{m=1}^{M^t}\sum_{k=1}^{K_m^t} H^t(m,k)$, is

$$\sum_{m=1}^{M^t}\sum_{k=1}^{K_m^t}\frac{q(t)a_{m,k}^t}{\sum_{m=1}^{M^t}\sum_{k=1}^{K_m^t}} + V[G_m^t(a_{m,k}^t) - \sum_{j=1}^{|\mathcal{J}^t|}\sum_{n=1}^{|\mathcal{N}_j^t|} C_{m,k,n}], \tag{8}$$

where $C_{m,k,n}=\omega_n(\mu^t_{m,n}x^t_{m,k,n}f^t_{m,k}+\gamma^t_{m,n}\alpha(x^t_{m,k,n}r_n)^2 f^t_{m,k})$. Eq. (8) implies that when the worker selection is fixed, we can determine $f^t_{m,k}$ and maximize the value of $H^t(m,k)$ for each subtask $s^t_{m,k}$ independently. In other words, we add up the maximized values of $H^t(m,k)$ for each subtask $s^t_{m,k}$ to obtain the maximum of objective function in problem $\mathbb{P}_2$. Based on the above insight, we define the set $\boldsymbol{\mathcal{F}}^t_{m,k}=\{f^t_{m,k}|\partial H^t(m,k)/\partial f^t_{m,k}=0\}$, which means that the first-order derivative of $H^t(m,k)$ with respect to $f^t_{m,k}$ in $\boldsymbol{\mathcal{F}}^t_{m,k}$ equals 0. When the worker selection is fixed, we get the optimal frame rate decision for each subtask $s^t_{m,k}$ as

$$f^t_{m,k}=\underset{f^t_{m,k}\in\boldsymbol{\mathcal{F}}^t_{m,k}\cup\{1,F^t_m\}}{\arg\max} H^t(m,k). \tag{9}$$

The crowdsourcing platform has to learn the optimal worker selection and frame rate adaptation on the fly. We then define a function $z^t_{m,k}$ to denote the value of $H^t(m,k)$ when the subtask $s^t_{m,k}$ is allocated to worker n and the frame rate is set to $f^t_{m,k}$. Thus, we have the following definition:

$$z^t_{m,k}=q(t)\epsilon^t_m(r_n)\phi^t_m(f^t_{m,k})/\sum\nolimits_{m=1}^{M^t}\sum\nolimits_{k=1}^{K^t_m} +V[G^t_m(a^t_{m,k})-\omega_n(\mu^t_{m,n}f^t_{m,k}+C^d_n\gamma^t_{m,n}\alpha(r_n)^2 f^t_{m,k})]. \tag{10}$$

According to Eq. (10), we notice that $z^t_{m,k}(n,f^t_{m,k})$ relies on the values of the energy consumption rates $\mu^t_{m,n}$ and $\gamma^t_{m,n}$.

Therefore, the **Algorithm 1** prepares such function for the volatile multi-armed bandit, as shown in line 6. The function clarifies the form in terms of these pending inputs, and the volatile multi-armed bandit uses its maintained inputs to make the decisions. After receiving the feedback from the volatile multi-armed bandit, **Algorithm 1** updates the queue backlog for next time slot, as shown in line 8 and decoupling.

B. Dispatch upon Volatile Multi-armed Bandit

Besides the fact that $\gamma^t_{m,n}$ and $\mu^t_{m,n}$ are stochastic, the varying set of candidate workers also poses a big challenge when learning the optimal worker selection and frame rate adaptation (i.e., the solution to $\mathbb{P}_2$). The crowdsourcing platform needs to restart the learning process when some worker quits or joins. This learning strategy is inefficient because it simply restarts the learning process without reusing what has been learned. Although some workers may quit or join the available worker set, the information of other workers still remain the same. Thus, we need to propose a learning algorithm which effectively reuses the already learned information.

In order to efficiently learn the optimal worker in a varying worker set, we adopt the volatile MAB framework [46], where workers may unexpectedly quit or join with unknown lifespan. We define the concept of the epoch as an interval in which candidate worker set is invariant. We let J^t to represent the total number of epochs in each time slot t, which is unknown in advance. When allocating the subtasks of task s^t_m, the lifespan for each worker n is denoted as $[u^t_{m,n},v^t_{m,n}]$ with $1\le u^t_{m,n}\le v^t_{m,n}\le K^t_m$, which indicates that worker n is present from the subtask $u^t_{m,n}$ through the subtask $v^t_{m,n}$. Assume that each worker only joins once during each task. If a worker joins for the second time, it will be treated as a new worker.

Algorithm 2: Dispatch upon Volatile MAB

Input: F^t_m, A^{min}, ω_n, G^t_m, $\forall t$

1 Remove $s^t_{m,k}$ out of $\boldsymbol{\mathcal{L}}^t_m$;
2 **if** $\exists$ *New Worker* $\bar{n}\in\boldsymbol{\mathcal{N}}^t_j$ **then**
3 $\quad u^t_{m,\bar{n}}=K^t_m-|\boldsymbol{\mathcal{L}}^t_m|$;
4 $\quad$ Send $s^t_{m,k}$ to $\bar{n}$, observe $\tilde{\mu}^t_{m,\bar{n}}$, $\tilde{\gamma}^t_{m,\bar{n}}$;
5 $\quad\bar{\mu}^t_{m,\bar{n}}\leftarrow\tilde{\mu}^t_{m,\bar{n}}$, $\bar{\gamma}^t_{m,\bar{n}}\leftarrow\tilde{\gamma}^t_{m,\bar{n}}$, $\theta^t_{m,\bar{n}}\leftarrow 1$;
6 **else**
7 $\quad\bar{n},\bar{f}\leftarrow\text{argmax}\{z^t_{m,k}(\bar{\mu}^t_{m,n},\bar{\gamma}^t_{m,n})+\sqrt{\frac{2\ln u^t_{m,n}}{\theta^t_{m,n}}}\}$;
8 $\quad$ Send $s^t_{m,k}$ to worker $\bar{n}$, observe $\tilde{\mu}^t_{m,\bar{n}}$, $\tilde{\gamma}^t_{m,\bar{n}}$;
9 $\quad\bar{\mu}^t_{m,\bar{n}}\leftarrow\frac{\bar{\mu}^t_{m,\bar{n}}\theta^t_{m,\bar{n}}+\tilde{\mu}^t_{m,\bar{n}}}{\theta^t_{m,\bar{n}}+1}$, $\bar{\gamma}^t_{m,\bar{n}}\leftarrow\frac{\bar{\gamma}^t_{m,\bar{n}}\theta^t_{m,\bar{n}}+\tilde{\gamma}^t_{m,\bar{n}}}{\theta^t_{m,\bar{n}}+1}$;
10 $\quad\theta^t_{m,\bar{n}}\leftarrow\theta^t_{m,\bar{n}}+1$;

Output: $z^t_{m,k}(\bar{\mu}^t_{m,\bar{n}},\bar{\gamma}^t_{m,\bar{n}})$

Algorithm 2 dispatches the video subtasks upon the volatile multi-armed bandit, which leverages the bandit to maintain an upper confidence bound on the empirical estimation of $z^t_{m,k}$ for each worker. It updates the estimation of all candidate workers as more subtasks have been allocated to them, and then the next subtask will be allocated to the worker as

$$\bar{n},\bar{f}\leftarrow\text{argmax}\{z^t_{m,k}(\bar{\mu}^t_{m,n},\bar{\gamma}^t_{m,n})+\sqrt{\frac{2\ln u^t_{m,n}}{\theta^t_{m,n}}}\}. \tag{11}$$

For each task s^t_m, the crowdsourcing platform allocates one subtask to every worker n, determining the frame rate $f^t_{m,k}$ according to Eq. (9) based on observed values $\tilde{\mu}^t_{m,n}$ and $\tilde{\gamma}^t_{m,n}$. If the observations are accurate, namely $\tilde{\mu}^t_{m,n}=\mu^t_{m,n}$ and $\tilde{\gamma}^t_{m,n}=\gamma^t_{m,n}$ (and hence $\tilde{z}^t_{m,k}=z^t_{m,k}$), then learning process can be terminated. And for the remaining subtasks of task s^t_m, the crowdsourcing platform will select the optimal worker n and frame rate $f^t_{m,k}$, the solution to $\max_{n,f^t_{m,k}} z^t_{m,k}$. However, due to the variance in energy consumption $\tilde{\mu}^t_{m,n}$ and $\tilde{\gamma}^t_{m,n}$, $\tilde{z}^t_{m,k}$ is only a noisy version of $z^t_{m,k}$. Thus, this simple learning scheme will perform very poorly because the allocated subtasks may get trapped in some worker whose $z^t_{m,k}$ is actually small. Therefore, a more sophisticated and effective learning algorithm requires continuous learning to smooth out the measurement noise. Actually, selecting the optimal worker and frame rate for subtasks of each subtask manifests a sequential decision making problem, which involves the critical tradeoff between exploration and exploitation: on one hand, the crowdsourcing platform needs to explore different workers by allocating subtasks to them and learning good estimates of $z^t_{m,k}$; on the other hand, it tends to allocate as many subtasks as possible to a priori unknown optimal worker and further select the optimal frame rate for subtasks.

For each epoch, **Algorithm 2** is designed to learn the optimal worker and frame rate. In lines 2-5, the initialization only applies to the new workers in each epoch, and the information for the remaining workers is retained and hence reused. Lines 6-10 are the continuous learning phase for each epoch, to find a priori unknown optimal worker, taking into account the joining time of the worker, i.e., line 7.

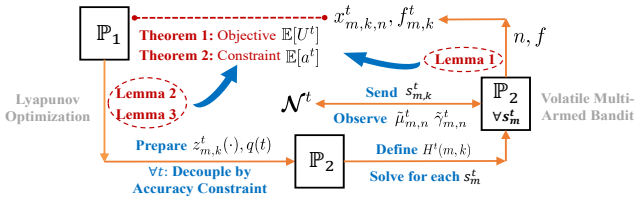


Fig. 6. Relationships between Lemmas and Theorems

IV. Performance Analysis

The relationships between all the lemmas and theorems are illustrated in Fig. 6, where the main result regarding the regret of long-term objective is shown in **Theorem 1** and the result regarding the long-term constraint is shown in **Theorem 2**.

Preliminaries: When using the volatile multi-armed bandit, the regret we use for s^t_m is defined in details as follows:

$$R_m(t) = \sum_{k=1}^{K^t_m} \mathbb{E}[z^t_{m,k}(n, f^t_{m,k}) - z^t_{m,k}(n^{t,*}_m, f^{t,*}_{m,k})],$$

where $z^t_{m,k}(n, f^t_{m,k})$ is the objective in $\mathbb{P}_2$ for subtask $s^t_{m,k}$ by **Algorithm 2**, and $z^t_{m,k}(n^{t,*}_m, f^{t,*}_{m,k})$ is achieved by selecting the optimal worker $n^{t,*}_m$ and frame rate $f^{t,*}_{m,k}$ for subtask $s^t_{m,k}$ that solve problem $\mathbb{P}_2$ when all of the inputs are observed.

Lemma 1. *For slot t, when adopting **Algorithm 2** to approximately solve problem $\mathbb{P}_2$ without priori information $\mu^t_{m,n}$ and $\gamma^t_{m,n}$, the regret for each task s^t_m is upper bounded as follows:*

$$R_m(t) \leq |\mathcal{J}^t| \sum_{n \neq n^{t,*}_m} 8ln(K^t_m)/\Delta^t_{m,n} + 8\Delta^t_{m,n}/3,$$

where $n^{t,*}_m$ is the optimal worker selected for task s^t_m, and $\Delta^t_{m,n}$ is defined as $\mathbb{E}[z^t_{m,k}(n^{t,*}_m, f^{t,*}_{m,k})] - \mathbb{E}[z^t_{m,k}(n, f^t_{m,k})]$.

Proof. See Appendix A, via volatile multi-armed bandit. □

Lemma 2. *When $\mathbb{P}_2$ is solved with the optimal solution, the time-averaged crowdsourcing profit for $\mathbb{P}_1$ satisfies:*

$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}[U^t] \geq p^{opt} - B/V,$$

where p^{opt} is the optimal crowdsourcing profit in $\mathbb{P}_1$ that can be obtained by ignoring the accuracy constraint.

Proof. See Appendix B, by using Lyapunov optimization. □

Lemma 3. *When the subproblem $\mathbb{P}_2$ is solved with the optimal solution, the analysis accuracy for $\mathbb{P}_1$ satisfies:*

$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}[a^t] \geq A^{min} + \frac{1}{\varepsilon}[B + V(p^{opt} - p^{min})],$$

where p^{min} is the objective of the worst solution for problem $\mathbb{P}_1$, and $\varepsilon < 0$ is a constant as the long-term accuracy surplus achieved by some stationary control policy.

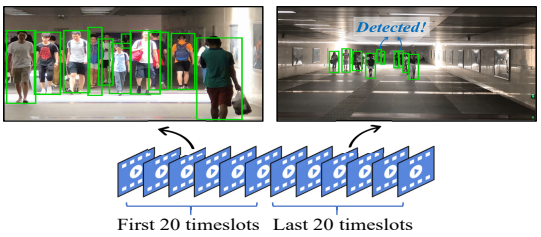
Proof. See Appendix C, also via Lyapunov optimization. □

Theorem 1. *The time-averaged crowdsourcing profit achieved by adopting **Algorithm 1** satisfies the following inequality:*

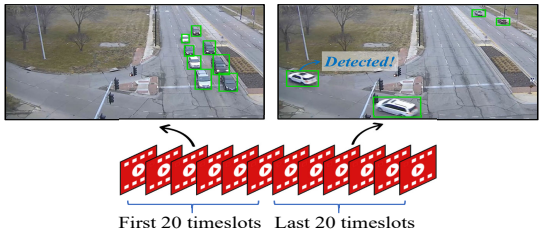
$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}[U^t] \geq p^{opt} - (B+W)/V,$$

where p^{opt} is the optimal profit and $\mathbb{P}_2$ is solved with a bounded deviation, denoted as the parameter W.

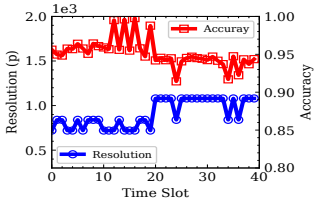
Proof. See Appendix D, combining Lemmas 1 and 2. □



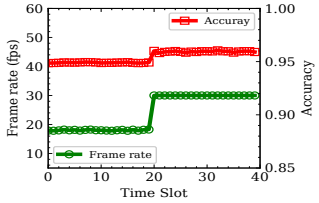
(a) Sequence of Tasks $s^t_1, \forall t \in \{1, ..., 40\}$



(b) Sequence of Tasks $s^t_2, \forall t \in \{1, ..., 40\}$



(c) Resolution and Accuracy



(d) Frame Rate and Accuracy

Fig. 7. Results of LOL in a demo example with $M^t = 2, \forall t$

Theorem 2. *The accuracy achieved by **Algorithm 1** satisfies:*

$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[a^t] \geq A^{min} + \frac{B+W+V(p^{opt}-p^{min})}{\varepsilon},$$

where p^{min} is the objective of the worst solution for problem $\mathbb{P}_1$, and $\varepsilon < 0$ is a constant as the long-term accuracy surplus achieved by some stationary control policy.

Proof. See Appendix E, combining Lemmas 1 and 3. □

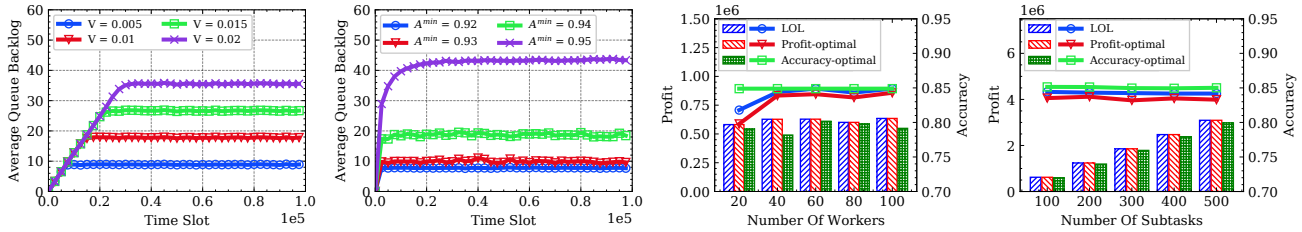
The above theorems demonstrate an $[O(1/V), O(V)]$ profit-accuracy tradeoff. Upon the trade-off, we can choose an arbitrarily large value of $V \to +\infty$ to drive the time-averaged crowdsourcing profit arbitrarily close to the optimal p^{opt} at a cost. Besides, **Theorem 2** also implies that the time-averaged accuracy queue backlog grows linearly with V.

V. Experiments and Result Analysis

In this section, we evaluate the performance of our proposed algorithm by extensive trace-driven experiments, and compare it against alternative algorithms to show the effectiveness.

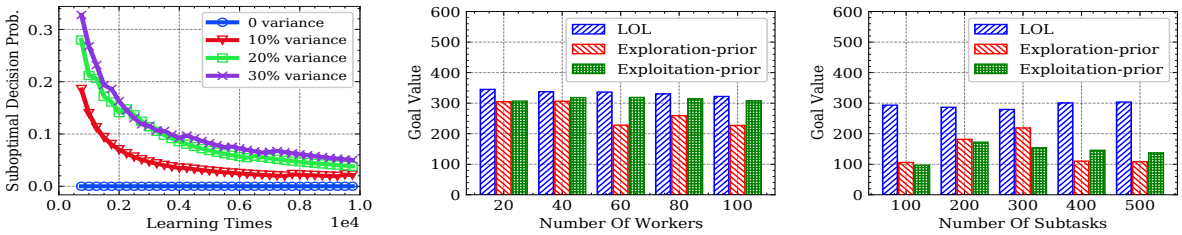
A. Experiment Settings

Our extensive trace-driven experiments use the videos derived from the AI City Datasets 2019 [36] for evaluating our online dispatcher for video analytics upon YOLOv3 [11], compared with other algorithms. The input frame sizes include 360p, 540p, 720p, 840p, 960p and 1080p, and the original frame rate is set as 30fps. Based on the traces derived from [14, 21, 22], we set the computation energy consumption



(a) Impact of V (b) Impact of A^{min} (c) Impact of Workers (d) Impact of Subtasks

Fig. 8. Impact of V and A^{min} on Average Queue Backlog, and Impact of Workers and Subtasks on Profit and Accuracy



(a) Suboptimal Decision Probability (b) Impact of Workers (c) Impact of Subtasks

Fig. 9. Suboptimal Decision and Impact of Workers and Subtasks

$\mu^t_{m,n} \sim N(5, 0.5)$ J/frame and the transmission energy consumption $\gamma^t_{m,n} \sim N(5, 0.5)\times 10^{-6}$ (J) by default. We generate the changes of the available worker set upon the dataset of Users Active Time Prediction [47]. Furthermore, we set the parameters $\alpha = 1$ and $\omega_n \sim U(0, 1)$ in the experiments.

We compare our proposed algorithms with 4 alternatives:

- **Profit-optimal** determines the worker selection and configuration for maximum profit, but it ignores the accuracy constraint, promised by the crowdsourcing platform.
- **Accuracy-optimal** maximizes the accuracy per slot for video analytics, but it ignores the crowdsourcing profit.
- **Exploration-prior** also conducts the learning of energy consumption rates for workers, but it prefers to perform the strategy of exploration in the bandit.
- **Exploitation-prior** also conducts the learning of energy consumption rates for workers, but it prefers to perform the strategy of exploitation in the bandit.

Except for these algorithms compared with LOL, LOL-M means Algorithm 1 triggers the traditional bandit.

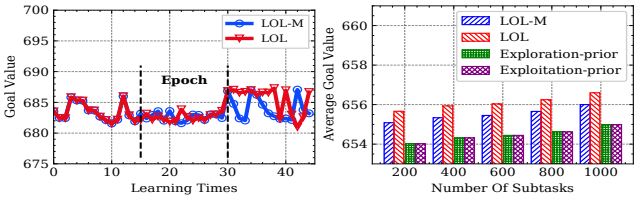
B. Experiment Results

Effectiveness of LOL: Figure 7 shows the results from LOL in a demo example with $M^t = 2, \forall t \in \{1, 2, ..., 40\}$. As demonstrated in Figure 7(a), the video analytics tasks in the first 20 time slots mainly contain the targets closed to the camera, while the targets in the last 20 time slots are faraway. In order to detect the targets far away and maintain high accuracy, the resolution is adapted to higher values when the crowdsourcing platform is selecting the configuration for the tasks in the last 20 time slots, as shown in Figure 7(c). Similarly, we learn from Figure 7(b) that the target velocities in the analytics tasks of the first 20 time slots and the last 20 time slots are different. Thus, a higher frame rate is needed to capture the targets of high speed and keep the analytics accuracy high in the last 20 time slots, which we can observe in Figure 7(d). The intuitions behind these adjustments are that "resolution can be reduced when the target size is large enough" and "some redundant frames can be skipped when the difference between the adjacent frames is small".

Accuracy Constraint: Figure 8 shows the impact of control parameter V as well as accuracy guarantee A^{min} on the average queue backlog. Each accuracy queue backlog gradually converges to a certain value. Thus, the long-term accuracy constraint can be satisfied if there are enough time slots. In Figure 8(a), we observe that when the value of V is large, the queue backlog needs more time slots to converge; when V decreases, the accuracy is the primary goal, and the accuracy queue easily becomes stable. Figure 8(b) shows the average queue backlog under different values of A^{min}. When A^{min} is high, the accuracy constraint can be violated, and then the convergency value of queue backlog will be large. On the contrary, a smaller A^{min} makes it easier to satisfy the accuracy constraint, thus keeping the queue backlog at a smaller value.

Profit: Figure 8 compares the profit and accuracy over the different numbers of workers/subtasks of LOL with other 2 benchmark algorithms. With the increase of the number of workers and subtasks, the LOL algorithm can still achieve good performance in profit and accuracy. It is worth noting that compared with the Profit-optimal algorithm, our proposed LOL algorithm can achieve almost the same profit and at the same time, LOL also obtains a higher accuracy.

Various Scenarios: We explore the changes of suboptimal decision probability as the learning times increases. Figure 9(a) shows that as the learning times of LOL increases, the probability of selecting suboptimal workers and configuration is reduced drastically. When the observation variance is zero, the crowdsourcing platform can always find the optimal worker after allocating one subtask to each available worker. When the observation variance increases, the probability of selecting suboptimal workers decreases. For the proposed LOL and other 2 algorithms, the impact of number of workers/subtasks on goal value (objective in $\mathbb{P}_2$) is shown in



(a) Objective of $\mathbb{P}_2$ (b) Impact of Subtasks

Fig. 10. LOL algorithm vs. LOL-M algorithm.

Figures 9(b) and 9(c). With the different number of workers and subtasks, LOL has a significant advantage in achieving higher goal value, which is improved by 68% on average.

We compare the proposed LOL algorithm with LOL-M under the varying worker set. We show the results by dividing the process of selecting workers and allocating subtasks into 3 epochs. At the beginning of epoch 2 (when learning times is 15), the optimal worker 1 in epoch 1 exits, while at the beginning of epoch 3 (learning times is 30), a new optimal worker 2 in epoch 3 joins. As shown in Figure 10(a), when the optimal worker 2 joins at the beginning of epoch 3, LOL-M requires some time to learn the information of all workers, which results in low goal values. However, for LOL, there is no need to learn the information again, and the high goal values can be maintained. LOL can retain the information of remaining workers while LOL-M restarts the learning process when there appears a change of the worker set. Furthermore, the goal values achieved by LOL-M, LOL and other 2 algorithms are demonstrated in Figure 10(b). Since LOL leverages the volatile multi-armed bandit to capture the workers' availability and the stochastic uses of worker resources, it outperforms the other 3 algorithms.

VI. Related Work

We summarize the prior studies by the following categories and highlight their shortcomings compared with our work.

A. Configuration for Video Analytics

Chameleon [9] was presented as a controller that dynamically picked the best configurations for NN-based video analytics pipelines. Distream [19] was proposed as a distributed live video analytics system, which was able to adapt to the workload dynamics to achieve low-latency, high-throughput, and scalable live video analytics. Wang *et al.* [14] developed an efficient online algorithm, which jointly optimized configuration adaption and bandwidth allocation in edge-based video analytics systems. Reducto [20] was built to dynamically adapt filtering decisions upon the time-varying correlation between feature type, filtering threshold, query accuracy and the video content. DDS [16] was presented to continuously send a low-quality video stream to the server, which executed the DNN with low latency to increase the inference accuracy.

However, these works fail to consider the crowdsourcing profit when conducting the video analytics over workers.

B. Selection for Crowdsourcing

Wang *et al.* [26] focused on the insufficient participation problem with limited number of workers and proposed to leverage social network to recruit workers as well as expanding the worker pool. Gao *et al.* [27] focused on the unknown worker recruitment problem in mobile crowdsensing and determined a recruiting strategy to maximize the completion quality. Lu *et al.* [28] studied the problem of least workers selection to make crowdsourcing system perform sensing tasks more effective with certain constraints satisfied. Liu *et al.* [29] proposed a user recruitment strategy with truthful pricing to tackle the online recruitment problem constrained by a budget. Li *et al.* [30] studied how to select optimal mobile users to construct an accurate monitoring map under a limited budget.

These works focus on worker recruitment for crowdsourcing in different scenarios. However, almost no work considers the varying worker set along with the analytics for videos.

C. Changes on Resource Usage

Li *et al.* [31] proposed related online framework to solve the multi-dimensional large-scale task assignment for utility maximization by running atomic tasks in parallel on workers. Kang *et al.* [32] developed a multi-armed bandit framework to learn a worker's preferences and his reliabilities for different categories of tasks. Zhao *et al.* [33] tackled the unknown worker recruitment by modeling it as a differentially private multi-armed bandit game by seeing each worker as an arm the task completion quality as the reward of pulling arm. Jin *et al.* [35] studied the anytime batched multi-armed bandit problem and proposed an algorithm that achieves the asymptotically optimal regret for exponential families of reward distributions.

These works have already treated the stochastic usage of resources, but fail to adjust the configurations for videos to maximize the profit for crowdsourcing platform.

VII. Conclusion

To maximize the crowdsourcing profit for video analytics, we formulate a non-linear integer program, considering both resource-quality trade-off for videos and workers' dynamic availability. We then propose a series of subproblems for tackling the proposed time-coupled long-term constraint, upon the Lyapunov optimization and volatile multi-armed bandit. Essentially, each one of the subproblems decides the configurations and workers for videos upon the prediction calculated by the bandit. Via rigorous proof, the regret measuring the gap between online decisions and the optimum is ensured. Extensive trace-driven experiments show the improvement of our proposed algorithm LOL compared with other algorithms.

Appendix

A. Proof of Lemma 1

Proof. To prove Lemma 1, we first introduce a proposition:

Proposition 1. *(Hoeffding) Given independent random variables where $a_i \le X_i \le b_i$, almost surely (with prob. 1) we have*

$$\mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} X_i - \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[X_i] \ge \varepsilon\right) \le \exp\left(\frac{-2\varepsilon^2 m^2}{\sum_{i=1}^{m}(b_i-a_i)^2}\right). \quad (12)$$

Proof. The proof can be obtained from [48]. □

In epoch j of slot t, when the platform is allocating the subtasks of s_m^t, we define the optimal worker n as $n_{m,j}^{t,*}$, which maximize $\mathbb{E}[z_{m,k}^t(n, f_{m,k}^t)]$. When there is no conflict of meaning, we abbreviate $n_{m,j}^{t,*}$ as n^*. Then worker $n \neq n^*$ will only be selected in the 2 cases: workers n^* and n have been explored insufficiently to distinguish between their means, or the upper confidence bound obtained from Hoeffding's inequality fails for either worker n^* or n. We consider the first case and bound the chance that a suboptimal worker will be selected due to insufficient sampling.

Let $z_{m,n}^t$ denote $\mathbb{E}[z_{m,k}^t(n, f_{m,k}^t)]$ and suppose there are two events: A is defined as $\bar{z}_{m,n}^t \leq z_{m,n}^t + \sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}}$, and B is $\bar{z}_{m,n^*}^t \geq z_{m,n^*}^t - \sqrt{\frac{2\ln u_{m,n^*}^t}{\theta_{m,n^*}^t}}$. In order to bound the probability of the complementary events of A and B occurring, we apply the Hoeffding's Inequality in Proposition 1. Since event A fails when $\bar{z}_{m,n}^t - z_{m,n}^t > \sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}}$, we plug $\varepsilon = \sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}}$, and get $\mathbb{P}(\bar{z}_{m,n}^t - z_{m,n}^t > \sqrt{(2\ln u_{m,n}^t)/\theta_{m,n}^t}) \leq {u_{m,n}^t}^{-4}$. Similarly, we have $\mathbb{P}(z_{m,n^*}^t - \bar{z}_{m,n^*}^t) > \sqrt{(2\ln u_{m,n^*}^t)/\theta_{m,n^*}^t}) \leq {u_{m,n^*}^t}^{-4}$.

Next, we try to bound the number of suboptimal worker selection in the epoch j of time slot t. A suboptimal worker n is only selected if its upper confidence bound exceeds that of worker n^*, i.e., $\bar{z}_{m,n}^t + \sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}} > \bar{z}_{m,n^*}^t + \sqrt{\frac{2\ln u_{m,n^*}^t}{\theta_{m,n^*}^t}}$. If A is true, we have $\bar{z}_{m,n}^t + \sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}} \leq z_{m,n}^t + 2\sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}}$. Similarly for event B, we have $\bar{z}_{m,n^*}^t + \sqrt{\frac{2\ln u_{m,n^*}^t}{\theta_{m,n^*}^t}} \geq z_{m,n^*}^t$. Combining the above 3 inequalities, we get $z_{m,n}^t + 2\sqrt{\frac{2\ln u_{m,n}^t}{\theta_{m,n}^t}} > z_{m,n^*}^t$. Calculating the number of times $\theta_{m,n}^t$ that worker n has been selected in the epoch j of time slot t, we get $\theta_{m,n}^t < 8\Delta_n^{-2}\ln u_{m,n}^t \leq 8\Delta_n^{-2}\ln K_m^t$, where $\Delta_n = \max_{n'}\{z_{m,n'}^t - z_{m,n}^t\}$. Thus, if both of these two events A and B hold, we select worker n at most $8\Delta_n^{-2}\ln K_m^t$ times.

Recall that worker $n \neq n^*$ will only be selected if either it is sampled insufficiently (fewer than $8\Delta_n^{-2}\ln K_m^t$), or event A or B fails. For worker n, we calculate the expected number of times it is selected in the epoch j of time slot t as

$$\mathbb{E}[\theta_{m,n}^t] \leq 8\Delta_n^{-2}\ln K_m^t + \sum_{k=1}^{K_m^t}\mathbb{E}[\mathbb{I}\{\bar{A}\cup\bar{B}\}] \tag{13a}$$

$$\leq 8\Delta_n^{-2}\ln K_m^t + \sum_{k=1}^{K_m^t}\mathbb{E}[\mathbb{I}\{\bar{A}\}] + \mathbb{E}[\mathbb{I}\{\bar{B}\}] \tag{13b}$$

$$\leq 8\Delta_n^{-2}\ln K_m^t + 8/3, \tag{13c}$$

where the inequality (13c) follows that for any k', $\sum_{k'=1}^{K_m^t} k'^{-4} \leq 1 + \int_1^\infty x^{-4}dx = 1 + \frac{-1}{1-4} = 4/3$. Since there are J^t epochs in time slot t, the regret is proved. □

B. Proof of Lemma 2

Proof. To prove Lemma 2, we first introduce a proposition:

Proposition 2. *For any $\delta > 0$, there exists a stationary and randomized policy Π for $\mathbb{P}_1$, which decides $x^{\Pi,t}$ and $f^{\Pi,t}$ independent of the current queue backlog $q(t)$, such that the following inequalities are satisfied:*

$$\mathbb{E}[A^{min} - a^{\Pi,t}] \leq \delta, \text{ and } \mathbb{E}[I^{\Pi,t} - O^{\Pi,t}] \geq p^{opt} - \delta. \tag{14}$$

Proof. The proof can be obtained by Theorem 4.5 in [49], which is omitted for brevity. □

Assuming we can get the optimal solution to $\mathbb{P}_2$, The strategies that maximize the utility in $\mathbb{P}_1$ among feasible decisions include the policy Π in Proposition 2. By plugging Proposition 2 into the drift-plus-penalty inequality, we have

$$\Delta(q(t)) - V\mathbb{E}[U^t|q(t)]$$
$$\leq B + q(t)\mathbb{E}[A^{min} - a^{\Pi,t}|q(t)] - V\mathbb{E}[U^{\Pi,t}|q(t)] \tag{15a}$$
$$\leq B + \delta q(t) - V(p^{opt} - \delta). \tag{15b}$$

By letting δ approach 0, summing the above inequality over $t \in \{0, 1, ..., T\}$ and dividing the result by T, we obtain

$$\frac{\mathbb{E}[L(q(t+1)) - L(q(0))] - V\sum_{t=0}^T \mathbb{E}[U^t|q(t)]}{T} \leq B - Vp^{opt}. \tag{16}$$

Considering that $L(q(0)) = 0$ and $L(q(t)) \geq 0$, we rearrange the terms in (16) and get the time-averaged profit bound, i.e.,

$$\lim_{T\to+\infty}\frac{1}{T}\sum_{t=0}^T \mathbb{E}[U^t] \geq p^{opt} - B/V. \qquad \square$$

C. Proof of Lemma 3

Proof. To obtain time-averaged accuracy bound, we assume there exists $\varepsilon < 0$, $\Phi(\epsilon)$ and a policy Γ that satisfy

$$\mathbb{E}[A^{min} - a^{\Pi,t}] \leq \varepsilon, \text{ and } \mathbb{E}[U^{\Pi,t}] = \Phi(\varepsilon). \tag{17}$$

Plugging (17) into drift-plus-penalty inequality, we get

$$\Delta(q(t)) - V\mathbb{E}[U^t|q(t)] \leq B + \varepsilon q(t) - V\Phi(\varepsilon). \tag{18}$$

By summing the above inequality over $t \in \{0, 1, ..., T\}$ and rearranging the terms, we have

$$\frac{1}{T}\sum_{t=0}^T \mathbb{E}[q(t)] \leq \frac{B + V(\frac{1}{T}\sum_{t=1}^T \mathbb{E}[U^{\Pi,t}] - \Phi(\varepsilon))}{-\varepsilon}$$
$$\leq [B + V(p^{opt} - p^{min})]/(-\varepsilon). \tag{19}$$

Considering $\sum_{t=0}^T \mathbb{E}[q(t)] \geq \sum_{t=0}^T \mathbb{E}[A^{min} - a^t]$, we have

$$\frac{1}{T}\sum_{t=0}^T \mathbb{E}[a^t] \geq A^{min} + \frac{1}{\varepsilon}[B + V(p^{opt} - p^{min})]. \tag{20}$$

Taking a "lim" sup of the above inequality as $T \to +\infty$, we obtain the time-averaged accuracy bound. □

D. Proof of Theorem 1

Proof. Considering $\mathbb{P}_2$ can be approximately solved by Algorithm 2 within a bounded deviation W based on Lemma 1, and plugging Proposition 2 into drift-plus-penalty, we have

$$\Delta(q(t)) - V\mathbb{E}[U^t|q(t)] \leq B + \delta q(t) - V(p^{opt} - \delta) + W. \tag{21}$$

Then, based on the proof of Lemma 2, we get the profit bound, $\lim_{T\to+\infty}\frac{1}{T}\sum_{t=0}^T \mathbb{E}[U^t] \geq p^{opt} - (B + W)/V$. □

E. Proof of Theorem 2

Proof. Considering $\mathbb{P}_2$ can be approximately solved by Algorithm 2 within a bounded deviation W based on Lemma 1, and using the ε and Φ defined in Lemma 3, we get

$$\Delta(q(t)) - V\mathbb{E}[U^t|q(t)] \leq B + \varepsilon q(t) - V\Phi(\varepsilon) + W. \tag{22}$$

Then, based on the proof of Lemma 3, we have

$$\frac{1}{T}\sum_{t=0}^T \mathbb{E}[a^t] \geq A^{min} + \frac{1}{\varepsilon}[B + W + V(p^{opt} - p^{min})]. \tag{23}$$

Taking a "lim" sup of the above inequality as $T \to +\infty$, we obtain the time-averaged accuracy bound. □

REFERENCES

[1] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *IEEE INFOCOM*, 2019, pp. 1045–1053.

[2] Y. Wang, Y. Gao, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Elsevier CN*, vol. 171, pp. 107 144–107 157, 2020.

[3] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE IoTJ*, vol. 6, no. 5, pp. 8095–8113, 2019.

[4] Y. Tian, W. Wei, Q. Li, F. Xu, and S. Zhong, "Mobicrowd: Mobile crowdsourcing on location-based social networks," in *IEEE INFOCOM*, 2018, pp. 2726–2734.

[5] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM*, 2019, pp. 2053–2061.

[6] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation," *Elsevier CVIU*, vol. 131, pp. 88–99, 2015.

[7] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *ACM SIGMETRICS*, 2013, pp. 81–92.

[8] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," in *AAAI Workshop*, 2011.

[9] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *ACM SIGCOMM*, 2018, pp. 253–266.

[10] G. Raghavan, A. Salomaki, and R. Lencevicius, "Model based estimation and verification of mobile device performance," in *ACM EMSOFT*, 2004, pp. 34–43.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE CVPR*, 2016, pp. 779–788.

[12] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," 2013.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE CVPR*, 2017, pp. 2117–2125.

[14] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *IEEE INFOCOM*, 2020, pp. 257–266.

[15] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *IEEE CVPR*, 2016, pp. 4820–4828.

[16] K. Du, A. Pervaiz, X. Yuan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *ACM SIGCOMM*, 2020, pp. 557–570.

[17] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE TCSVT*, vol. 15, no. 5, pp. 645–658, 2005.

[18] C. Zhou, C.-K. Tham, and M. Motani, "Online auction for scheduling concurrent delay tolerant tasks in crowdsourcing systems," *Elsevier CN*, vol. 169, pp. 107 045–107 061, 2020.

[19] X. Zeng, B. Fang, H. Shen, and M. Zhang, "Distream: scaling live video analytics with workload-adaptive distributed edge intelligence," in *ACM SenSys*, 2020, pp. 409–421.

[20] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, "Reducto: On-camera filtering for resource-efficient real-time video analytics," in *ACM SIGCOMM*, 2020, pp. 359–376.

[21] H. Qian and D. Andresen, "Reducing mobile device energy consumption with computation offloading," in *IEEE SNPD*, 2015, pp. 1–8.

[22] H. Li and L. Chen, "Rssi-aware energy saving for large file downloading on smartphones," *IEEE LES*, vol. 7, no. 2, pp. 63–66, 2015.

[23] T. Elgamal, S. Shi, V. Gupta, R. Jana, and K. Nahrstedt, "Sieve: Semantically encoded video analytics on edge and cloud," in *IEEE ICDCS*, 2020, pp. 1383–1388.

[24] C. Neff, M. Mendieta, S. Mohan, M. Baharani, S. Rogers, and H. Tabkhi, "Revamp 2 t: Real-time edge video analytics for multicamera privacy-aware pedestrian tracking," *IEEE IoT-J*, vol. 7, no. 4, pp. 2591–2602, 2019.

[25] C. Qu, S. Wang, and P. Calyam, "Dycoco: A dynamic computation offloading and control framework for drone video analytics," in *IEEE ICNP*, 2019, pp. 1–2.

[26] Z. Wang, Y. Huang, X. Wang, J. Ren, Q. Wang, and L. Wu, "Socialrecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks," *IEEE TMC*, vol. 20, no. 5, pp. 2055–2066, 2021.

[27] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE INFOCOM*, 2020, pp. 179–188.

[28] Z. Lu, Y. Wang, Y. Li, X. Tong, C. Mu, and C. Yu, "Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial iot," *IEEE T IND INFORM*, 2021.

[29] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile crowdsensing," in *IEEE INFOCOM*, 2020, pp. 1113–1122.

[30] J. Li, J. Wu, and Y. Zhu, "Selecting optimal mobile users for long-term environmental monitoring by crowdsourcing," in *IEEE IWQoS*, 2019, pp. 1–10.

[31] Q. Li and L. Cai, "Online task scheduling with workers variabilities in crowdsourcing," *IEEE Access*, vol. 9, pp. 78 025–78 034, 2021.

[32] Q. Kang and W. P. Tay, "Task recommendation in crowdsourcing based on learning preferences and reliabilities," *IEEE T SERV COMPUT*, 2020.

[33] H. Zhao, M. Xiao, J. Wu, Y. Xu, H. Huang, and S. Zhang, "Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits," *IEEE TMC*, pp. 1–15, 2020.

[34] A. Nika, S. Elahi, and C. Tekin, "Contextual combinatorial volatile multi-armed bandit with adaptive discretization," in *PMLR AISTATS*, 2020, pp. 1486–1496.

[35] T. Jin, J. Tang, P. Xu, K. Huang, X. Xiao, and Q. Gu, "Almost optimal anytime algorithm for batched multi-armed bandits," in *PMLR ICML*, 2021, pp. 5065–5073.

[36] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu, "The 2019 ai city challenge," in *IEEE CVPR Workshops*, 2019, p. 452–460.

[37] Y. Sun, S. Zhou, and J. Xu, "Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE JSAC*, vol. 35, no. 11, pp. 2637–2646, 2017.

[38] Y. Chen, S. Zhang, M. Xiao, Z. Qian, J. Wu, and S. Lu, "Multi-user edge-assisted video analytics task offloading game based on deep reinforcement learning," in *IEEE ICPADS*, 2020, pp. 266–273.

[39] M. Hu, Z. Xie, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Heterogeneous edge offloading with incomplete information: A minority game approach," *IEEE TPDS*, vol. 31, no. 9, pp. 2139–2154, 2020.

[40] J. Dong and Y. Ye, "Adaptive downsampling for high-definition video coding," *IEEE TCSVT*, vol. 24, no. 3, pp. 480–488, 2014.

[41] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling," *IEEE TIP*, vol. 20, no. 11, pp. 3291–3296, 2011.

[42] Q. Liu and T. Han, "Dare: Dynamic adaptive mobile augmented reality with edge computing," in *IEEE ICNP*, 2018, pp. 1–11.

[43] Z. Lu, S. Rallapalli, K. Chan, and T. La Porta, "Modeling the resource requirements of convolutional neural networks on mobile devices," in *ACM MM*, 2017, pp. 1663–1671.

[44] P. Lai, Q. He, G. Cui, F. Chen, M. Abdelrazek, J. Grundy, J. Hosking, and Y. Yang, "Quality of experience-aware user allocation in edge computing systems: A potential game," in *IEEE ICDCS*, 2020, pp. 223–233.

[45] F. Liu, P. Shu, and J. C. Lui, "Appatp: An energy conserving adaptive mobile-cloud transmission protocol," *IEEE TC*, vol. 64, no. 11, pp. 3051–3063, 2015.

[46] Z. Bnaya, R. Puzis, R. Stern, and A. Felner, "Social network search as a volatile multi-armed bandit problem," *Human*, vol. 2, no. 2, pp. pp–84, 2013.

[47] "mobile usage time prediction," https://www.kaggle.com/bhuvanchennoju/mobile-usage-time-prediction, 2021.

[48] J. Duchi, "Probability bounds," 2009. [Online]. Available: https://stanford.edu/~jduchi/projects/probability_bounds.pdf

[49] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.